REMARKS

Claims 1 and 4-14 were examined by the Office, and in the Office Action of September 21, 2009 all claims are rejected. With this response claim 5 is cancelled without prejudice. Applicant respectfully requests reconsideration and withdrawal of the rejections in view of the following discussion.

Claim Rejections Under § 112

In section 5, on page 3 of the Office Action, claims 1 and 5 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. Applicant respectfully submits that the rejection to claims 1 and 5 is moot in view of the cancellation of claim 5.

Claim Rejections Under § 103

In section 6, on page 4 of the Office Action, claims 1, 4-14 are rejected under 35 U.S.C. § 103(a) as unpatentable over Grohoski (U.S. Appl. Publ. No. 2004/0225885) in view of Srinivasan et al. (U.S. Appl. Publ. No. 2004/0158742). Applicant respectfully submits that the cited references, alone or in combination, fail to disclose or suggest all of the limitations recited in claim 1. Claim 1 is amended to recites that the first logical interface is not accessible when data is being transferred in the second logical interface, and the cited references at least fail to disclose or suggest this limitation of claim 1. Furthermore, applicant respectfully submits that the cited references also fail to disclose or suggest that the configuration register is configured to receive mode setting instructions from a protected application.

In contrast to claim 1, in Grohoski the crypto co-processor (250) can be accessed through a set of hardware registers, and the crypto co-processor (250) can share memory access units with the main CPU in order to reduce duplicated hardware. See Grohoski paragraph [0056]. The Office appears to assert on page 3 of the Office Action that the common memory corresponds to the first and second logical interfaces as recited in claim 1. In Grohoski the CPU (205) identifies a packet as a crypto packet, then identifies any additional data required to execute the crypto packet, then identifies the additional data in the control queue, and then transfers the crypto packet to the crypto co-processor (250). See Grohoski paragraphs [0058]-[0061]. At time T_1 the crypto co-processor (250) receives the crypto packet and retrieves the corresponding control word, and at time T_2 the crypto co-processor updates the control word to

identify the crypto packet as being completed. See Grohoski paragraph [0062]. However, also between time T1 and time T2 if a subsequent packet is processed in the CPU (205) and identified as a crypto packet, the crypto packet is forwarded to the crypto processor (250) and a corresponding control word is forwarded to the control queue. See Grohoski paragraph [0063]. Accordingly, whenever the crypto co-processor (250) is able to access the memory and/or interface, the CPU (205) is also able to access the same. Therefore, Grohoski fails to disclose or suggest that the first logical interface is not accessible when data is being transferred in the second logical interface, as recited in claim 1.

The Office asserts on page 2 of the Office Action that the applicant has not provided any teaching within Grohoski that supports applicant's argument that both the CPU and the cryptoprocessor can access the memory at the same time in Grohoski. However, the paragraph relied upon by the Office to show that Grohoski teaches that the first logical interface is not accessible when data is transferred in the second logical interface specifically states that the crypto coprocessor (250) also allows crypto processing to overlap with execution of normal (i.e. noncryptographic) instructions. See Grohoski paragraph [0056]. Applicant respectfully submits that at least this discussion in Grohoski makes it apparent that both the CPU and crypto processor can access the memory at the same time. The Office also asserts that since the transfers are made through a shared memory then only one can access it at a time. However, applicant respectfully disagrees, since some memory such as a multi-ported memory provides more than one access path to its content, and allows the same bank of memory to be read and written simultaneously. See e.g. definition from www.yourdictionary.com/computer/multiported-memory (attached as Appendix A). However, claim 1 specifically recites that the first logical interface is not accessible when data is transferred in the second logical interface. Therefore for at least the reasons stated above, applicant respectfully submits that Grohoski does not disclose or suggest this limitation of claim 1.

Furthermore, on page 5 of the Office Action, the Office acknowledges that Grohoski fails to disclose a configuration register configured to receive mode setting instructions from a protected application, and relies upon Srinivasan for this teaching. However, Srinivasan also fails to disclose or suggest that the configuration register is configured to receive mode setting instructions from a protected application, as recited in claim 1. In contrast to claim 1, Srinivasan only discloses that in a step (216) the trusted server optionally verifies that the secure processor

(110) is authorized to receive application software from the trusted server. See Srinivasan paragraph [0105]. However, Srinivasan further states that the CPU operating in secure mode receives the application software or other additional instructions from the trusted server. See Srinivasan paragraph [0107]. If the CPU is already operating in a secure mode before the application software is received from the trusted server, then the application software cannot be considered to be a protected application that provides mode setting instructions to a configuration register, as recited in claim 1.

In contrast to the present application, in Srinivasan applications corresponding to the protected applications recited in claim 1 are defined as "secure code" and "secure boot loader code." See Srinivasan paragraph [0036]. These protected applications are not the equivalent to the "application software," which the Office asserts corresponds to the protected applications recited in claim 1. Srinivasan defines "application software" as a set of instructions or parameters capable of being executed or interpreted by a processor. See Srinivasan paragraph [0031]. Since both secure code and application software are defined in the Lexicography provided in Srinivasan, it implies that they are differentiated from each other. Srinivasan makes no mention that the application software is a protected application as mentioned in claim 1. Therefore, the section relied upon by the Office does not disclose a configuration register configured to receive mode setting instructions from a protected application, as recited in claim 1. Instead, these sections only disclose that the application software places parameters for a request for services in a set of selected registers, or performs an uncached read to a register. See Srinivasan paragraphs [0121] & [0127]. Even if the application software are considered to be a protected application, which applicant does not admit, the functions performed by the application software in Srinivasan do not correspond to providing mode setting instructions, as recited in claim 1.

Furthermore, while Srinivasan defines "secure code" and "secure boot loader code" to be interpretable or executable by the secure processor, and known to the secure processor to be trustable, the secure code and secure boot loader code do not provide mode setting instructions to a configuration register. Claim 1 recites that the configuration register is configured to receive mode setting instructions <u>from</u> a protected application, however even if the secure code and secure boot loader code are considered to correspond to the protected application, Srinivasan does not disclose a configuration register configured to receive mode setting instructions from

the secure code or the secure boot loader code. Instead, after power on of the secure processor (110) a reset signal (A170) is asserted that indicates that the secure processor (110) has been reset. See Srinivasan paragraph [0088]. As a result, the secure mode active signal (A160) is asserted and the CPU transfers execution control to the secure boot code (A115). The secure mode active signal (A160) indicates to the non-volatile memory that the CPU is allowed to access the secure boot code, execute its instruction, and read and write data using the security information (113). See Srinivasan paragraph [0089]. However, Srinivasan does not disclose or suggest that a configuration register receives mode setting instructions from a protected application, instead it appears that the reset signal (A170) is responsible for setting the secure processor (110). Therefore, for at least these reasons claim 1 is not disclosed or suggested by the cited references.

Independent claim 12 is amended in a manner similar to claim 1, and contains limitations similar to claim 1. Therefore, for at least the reasons discussed above in relation to claim 1, claim 12 is not disclosed or suggested by the cited references.

The dependent claims depending from the above mentioned independent claims are not disclosed or suggested by the cited references at least in view of their dependencies.

Conclusion

It is therefore respectfully submitted that the present application is in condition for allowance and such action is earnestly solicited. The undersigned authorizes the Commissioner to charge any fees required to submit this response to Deposit Account No. 23-0442.

Respectfully submitted,

Dated: 16 December 2009

Keith R. Obert
Attorney for Applicant
Registration No. 58,051

WARE, FRESSOLA, VAN DER SLUYS
& ADOLPHSON LLP
Bradford Green, Building Five
755 Main Street, P.O. Box 224
Monroe, CT 06468
Telephone: (203) 261-1234
Facsimile: (203) 261-5676
USPTO Customer No. 004955

APPENDIX A

multiported memory

multiported memory definition - computer

A type of memory that provides more than one access path to its contents. It allows the same bank of memory to be read and written simultaneously. See video RAM.

comments

Improve this definition.
Do you have more to add? Share your linguistic knowledge or observation.
Name: anonymous Login/Register to save your comments.

Comment:

Submit

Browse dictionary definitions near *multiported memory*

C multiport serial card
C multiport repeater
C multiport bridge
M multipolar neuron
Multiprise C
multiprocessing W
multiprocessing C
multiprocessor W

M multipolar cell

M multipolar

Multipoint Processor

C multipoint pairing

C multipoint line

Multipoint Controller

multiprocessor C

multiprogramming C

Multiprotocol Label Switching

Multiprotocol Lambda Switching

MultiProtocol over ATM

multiprotocol router C